E2OO 944

AD-A248 657

PL-TR-91-2248

Visual Photometric Experiment Data Processing System

Kevin P. Larson
Andrew J. Mazzella, Jr.

RDP Inc.
Waltham Office Center
486 Totten Pond Road
Waltham, MA 02154

DTIC
ELECTE
MAR 10, 1992
B
D

31 December 1991

Final Report
23 January 1987 thru 31 December 1991

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

PHILLIPS LABORATORY
AIR FORCE SYSTEMS COMMAND
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731-5000

92-06175

This technical report has been reviewed and is approved for publication

PAUL D. LeVAN
Contract Manager

STEPHAN D. PRICE
Branch Chief

ALAN D. BLACKBURN, COLONEL
Division Director

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION	1b RESTRICTIVE MARKINGS
Unclassified	

2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT
	Approved for public release;
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	Distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
	PL-TR-91-2248

6a NAME OF PERFORMING ORGANIZATION	6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION
RDP, Incorporated		Phillips Laboratory

6c ADDRESS (City, State, and ZIP Code)	7b ADDRESS (City, State, and ZIP Code)
486 Totten Pond Road	Hanscom Air Force Base
Waltham, Massachusetts 02154	Massachusetts 01731-5000

8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
		F19628-87-C-0065

8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
	62101F	7670	06	AS

11 TITLE (Include Security Classification)

Visual Photometric Experiment Data Processing System.

12 PERSONAL AUTHOR(S)
Larson, Kevin P., Mazzella, Andrew J., Jr.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final Report	FROM 87-Jan-23 TO 91-Dec-31	91-Dec-31	58

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Visual Radiometer; Zodiacal Light; Star Field Matching
17	03		
03	01		

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Data processing and analysis developments for the Visual Photometric Experiment (VIPER) are described in detail, including descriptions of data files and program execution.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Paul LeVan	(617) 377-4550	GOB

DD FORM 1473 84 MAR   83 APR edition may be used until exhausted
All other editions are obsolete

CONTENTS

ILLUSTRATIONS

iii

PREFACE

This report describes the software generated by the Data Analysis Services group of RDP Incorporated to perform the pre-flight and post-flight data processing and analysis for the VIPER mission. Other contributing members of this group are Michael Delorey, Peter Dickson, James Hughes, Steven Lacaire, John Palys, Sharon Poisson, and Rahul Rao.

The mission is being performed by PL/GPOB, PL/SXA, and their supporting contractors.

Introduction

The Visual Photometric Experiment (VIPER) Data Processing System is a series of programs and databases that bring the divergent data streams from the VIPER Get-Away Special Canister (GAS Can) to a unified form on a Digital Equipment Corporation (DEC) VAX computer system. Data streams originating from a visual radiometer, Xybion camera, and Pulnix camera in the GAS Can, NASA shuttle ephemeris tape, and Yale Bright Star Catalogue are unified by this software system to produce a spatial database. The spatial database, described in detail later in this report, is the foundation from which further zodiacal studies and investigations will be launched.

The VIPER Data Processing System can be divided into three general sections. The first section consists of the generation of Preprocessing databases. These databases are created from non-GAS Can data. The second section deals with transporting the recorded instrument data from the GAS Can's two distinct data recorders to the DEC/VAX computer system. The third section deals with the merging of the separate data sets from the first two sections into the spatial database.

Section I - Preprocessing Databases

As an initial step before the Data Processing System can be fully utilized, three databases must be generated from data not gathered from the GAS Can. These databases must be generated as inputs for the final process described in Section III can be executed but they may be generated simultaneously with the decommutation processing in Section II.

The first two databases are derived from the Yale Bright Star Catalogue and its Supplement[1]. The original Bright Star Catalogue magnetic tape is run through a series of programs to generate a precessed subset of the catalogue. The subset of sources is chosen on the basis of the magnitude of a source, with the brighter sources being selected.

Due to the spatial resolution of the VIPER cameras, the instruments would "see" only one source if two or more sources were within 20 arcseconds of each other. Therefore, the subset catalogue was processed to combine adjacent stars, and the resulting "coalesced" stars are stored in a FORTRAN Direct Access File (DAF). This process has effectively translated the Yale Bright Star Catalogue into a catalogue of stars as seen by the VIPER cameras.

---

[1] Hoffleit, D., and Jaschek, C., (1982) The BRIGHT STAR CATALOGUE, 4th Revised Edition, Yale University Observatory, New Haven, Connecticut

For purposes of attitude determination, a DAF is not the best way
to organize the data. When attempting to correlate a catalogue
entry with a camera image, a two-step method is needed. First, a
reference source in the camera image is compared to a Possible
Match Source (PMS) in the catalogue by magnitude. Then, the
neighbors of the reference source are compared to the PMS's
neighbors. The corresponding neighbors need to match in relative
distance from the PMS.

It would be advantageous to be able to access catalogue data by
magnitude and by nearest neighbors. VAX FORTRAN has this provision
by way of its Indexed Sequential Access Method (ISAM) files. ISAM
files provide for the use of key fields on each record by which a
particular record can be immediately accessed from the database
regardless of its position. Furthermore, after a single keyed-
access acquisition from the database, the database is then
accessible sequentially and the subsequent records will be acquired
in ascending order on that key field. The sorting is performed via
pointers stored in the file at the time of creation.

Two ISAM files are generated from the DAF. The first is a star-
coalesced ISAM file where each record has two keys by which it can
be accessed. This database is called the coalesced catalogue. The
keys are catalogue source number and coded magnitude value. The
catalogue source number key is unique to each source while the
coded magnit·Xxe key may not be unique. The coded magnitude key is
a value that includes the magnitude (M) and uncertainty in the
magnitude (DM), according to Equation 1:

$$Key = M \times 2560 + (DM \times 10 + 0.5)$$ (1)

This key is then truncated at the decimal point to provide an
integer magnitude value to be used as the coded magnitude key.
Other data for each record are limited to the celestial coordinates
of right ascension and declination. These are epoch 1950.0 values
precessed to the date of the flight with adjustments for proper
motion. Sources included in this database are selected according
to a magnitude threshold limit. Only sources with magnitudes less
than or equal to 6 are catalogued.

The second ISAM file contains the neighbors of each source in the
coalesced catalogue. A star's neighborhood is defined by a circle
with a radius of 13.5 degrees, where 13.5 degrees is the maximum
field-of-view of the Xybion camera, measured diagonally across the
video frame. This neighbor catalogue, referred to as a proximal
catalogue, has only one key for Indexed accessed. This key is
generated by taking the reference star's number (RSN) in the
coalesced catalogue, and the current neighbor's number, or CNN, and
defining the key according to Equation 2:

$$Key = RSN \times 100,000 + CNN$$ (2)

2

The RSN is defined so that the brightest star has an RSN of 1, the next brightest is 2, and so forth. Multiplying the RSN by 100,000 is done simply to shift the digits five decimal places allowing a non-overlapping concatenation of the RSN and the CNN. The CNN, like the RSN, is the ordinal number of the source in the coalesced catalogue. The CNN is never equal to the RSN. After accessing a record in the proximal catalogue, the key can be decoded to provide the CNN. Using the CNN, the magnitude and celestial coordinates of the neighbor can be obtained from the coalesced catalogue. There is no limit set for the maximum number of neighbors that a source can have. However, a neighbor must have a magnitude less than or equal to the magnitude cutoff used in the coalesced catalogue generation, namely 6. After a source has been completely processed for neighbors, a header record is written that uses Equation 1 for its key with the same RSN but the CNN is set to zero. Each record in the proximal catalogue has two data fields. The first data field is the key mentioned earlier. The value of the second field is dependent on the record type. For header record types, the second field specifies the number of neighbors of this source, while on the neighbor record types, this second field specifies the radial distance from the reference star measured in 10^i radians.

Figure 1 is a flowchart that indicates graphically the steps involved in processing the Yale Bright Star Catalogue to generate coalesced and proximal ISAM databases.

The last preprocessing database that needs to be generated is the CAS Database. The items that are included in the CAS data tape provided by NASA are selectable by the agency acquiring the information. The actual data format for the tape has not yet been obtained from NASA. Therefore, complete descriptions of the database and program that generates the database are not included in this report. A list is provided in Appendix A Section III of the chosen items to be included on the tape. FORTRAN null modules for data acquisition have been implemented in the programs where this data will be used for purposes of testing other aspects of the programs.

Appendix A provides a more detailed account of the data format of the output files associated with this processing section. All the software described in this section has been fully tested and has been demonstrated to function properly on the test data tapes generated by the VIPER instrument.

Section II - Flight Data Decommutation and Database Generation

A.   From VHS Recorder to VAX Mainframe (Star Cluster File)

Instrumentation on the VIPER GAS Can includes Pulnix and Xybion video cameras. The video outputs from the cameras are recorded on a flight-qualified, VHS format, video cassette recorder (VCR). The

data tape is a T-160 in length and recording is at the slowest possible speed, for approximately 2 hours and 40 minutes of video data. At any time, the data from only one camera are recorded but the cameras are alternately recorded every 12 seconds. Phillips Laboratory personnel will remove the VHS cassette from the GAS Can after the flight for commencement of the data playback process.

The playback hardware system consists of a standard VCR whose output is fed into a Data Translation Corporation's DT2851 Frame Grabber. The frame grabber is software-driven from a Zenith Z248 personal computer. The Z248 is an IBM-compatible, 80286-based machine with a math co-processor and monochrome monitor, utilizing DOS 3.1 and Microsoft FORTRAN Version 5. A television monitor is provided for displaying the VCR data. Data Translation provided the FORTRAN library of routines that enable user-written FORTRAN programs to drive the frame grabber board.

The program named FRAME has been written to acquire a single frame of video data. Keyboard input is required. The user is prompted for input parameters governing data display on the television monitor, time track generation mode (see description at the end of this paragraph), sampling rate and the duration of the program run (both are in real-time seconds). Camera definition values and calibration information are read from a file named SENSOR.SPC. A second input file, named INDEX.NUM, is read to provide information regarding bit placement of the time code relative to the sync pattern that precedes it. These inputs define bit width (measured in pixels), confidence limits, number of bits in the sync mark, number of bits in the time code, and the relative pixel number of each of the time code bits. VIPER encodes month, day number, hour, minute, second, and hundredths of a second of the controller clock, each as an 8 bit word. Appendix B provides a detailed description of how these values are defined. During VCR data processing, the program commands the frame grabber board to hold the current video frame in memory. At this point, all subsequent video input to the frame grabber board from the VCR is ignored until the acquisition of another frame is needed. The FRAME program then commands the frame grabber to transmit 32 rows of pixels to the PC. Included in the first set of 32 rows is the time code. The time code is a series of pixels on row 15 that are binary-coded-decimal (BCD) values of the time. The time code is decoded and saved for output. Thresholds for determining whether the pixel represents an "on" bit or an "off" bit are camera dependent. Active camera determination is performed by analyzing the peak pixel value in the time code. Because pixel values in the time code tend to be higher when the Xybion is active, the threshold for determining the active camera is a pixel value of 230 (on a scale of 0 through 255). If any of the first 6 pixels in the sync mark exceeds 230, the active camera is known to be the Xybion. The time code and active camera identification number are written to the output file PIXELS.VAL. If time track generation mode has been selected, the rest of this video frame is ignored and another frame is acquired. Otherwise,

4

processing continues as described below.

After time code determination, the full set of pixels that has been acquired is checked for possible stars. Pixel values are compared to a threshold to determine if the pixel is part of a star image. When a pixel is found to have a value above the threshold, the pixel's coordinates and value are written to the output file PIXELS.VAL and the search continues. After all 32 rows have been processed, another set of 32 rows is read and this procedure continues until all 16 sets of 32 rows have been processed. After completing this process, an end-of-frame record is written to the output file indicating that no more data exist for this video frame.

Program FRAME provides a significant video output that is selectable at the start of the program's execution, namely the option of selecting data display on the television monitor. Any video frame acquired for processing is simultaneously displayed. If the user selects the display of the visual radiometer's (VR) field-of-view (FOV), A circle will be displayed on the monitor indicating the extent of the FOV. Processing of video data will be slightly faster if data display is disabled. Figure 2 illustrates a sample data display on the television monitor. The data was recorded during a full VIPER test on 15 November 1990. This data is of the Pleiades as seen by the Xybion camera. The BCD time code is in the upper left hand corner in row 14 of the image. The VR FOV is shown as the oval above and to the left of the Pleiades. The distortion introduced by the monitor causes the circular FOV to appear as an oval. This data was displayed on the monitor and photographed for this report.

Though a file containing pixel values is now in existence, the data in the PIXELS.VAL file is not in a form that can be compared to catalogue sources. Early scenarios for the VIPER mission proposed scan rates up to one degree per second, performed as shuttle roll maneuvers about the local vertical axis. This scan rate would have produced blurring of star images across two or three pixels, which would have affected the determination of the position of the source and also spreads out the intensity of the source over more than one pixel. A follow-on program to FRAME was written to read the pixel information and to determine the star information it contains.

Program SDPROCSS runs on the Zenith 2248, described earlier, and performs the star discovery processing (SDPROCSS). The object of the star discovery processing algorithm is to associate pixels corresponding to the same object with each other, and to obtain position and luminosity information for each object, for use in the star field matching algorithm that is described later. The pixel file, PIXELS.VAL, is read one frame at a time. Clusters of pixels are determined that comprise one source. An intensity-weighted centroid is determined for each cluster and this intensity-weighted centroid is assumed to be the source's location in pixel

coordinates. By integrating over the cluster for source intensities and using calibration information, the source's magnitude can be evaluated. The calibration information is the same as that found in the file SENSOR.SPC, which is used by program FRAME. After all the clusters for a frame have been evaluated, the frame's header record, which was read from the PIXELS.VAL file, is written out in the same format, with the intensity-weighted centroids and the magnitudes of each source. These data are written to the file STARS.CLS. This file is uploaded to the PL/SC VAX-Cluster for later processing. Figure 3 describes this processing.

SDPROCSS can be transported to the VAX computer system with minimal effort required. This may be necessary if the star discovery processing becomes too time consuming on a PC. The data that will be transported will be larger but the increased processing speed of the VAX may easily offset this consideration.

The current mission scenario includes only relatively slow roll scans, with rates of about 8 arc-minutes per second. Thus, the optical image size of a few arc-seconds for point sources will only be marginally affected, and it may be possible to bypass the pixel-clustering process. Conversion from camera intensity to magnitude will still be required.

Appendix B provides file descriptions for all of the input and output files used in the processing of VCR data to their final form as a STARS.CLS file. Program FRAME has been fully tested. Program SDPROCSS has not been completely validated. Execution of the program provides adequate preliminary results, but conversion parameters from pixel intensity to corresponding star magnitude have not been determined. These values are currently being determined and will be incorporated into the software and tested by the flight of the VIPER payload.

B. From Sunstrand Data Recorder to VAX Mainframe (VR Data File)

In addition to the two video cameras, there is a visual radiometer, the primary VIPER sensor. Its output is recorded on a Sunstrand Data Recorder in the VIPER GAS Can. The high and low gain output voltages of the visual radiometer are sampled every one-hundredth of a second. The microprocessor that controls the VIPER GAS Can groups 5 sequential 12-bit samples from each gain type with various other GAS Can outputs to form a 240-bit minor frame of data. Twenty minor frames are grouped together to form one major frame of data that spans one second of time, resulting in one hundred VR samples. Section I of Appendix C illustrates the VIPER Format with a description of each item in the major and minor frames. Major frames are written to the Sunstrand Data Recorder's cartridge as part of a larger record. This cartridge will be removed by Phillips Laboratory personnel after the flight.

At Phillips Laboratory, a DEC PDP-11 computer running the RT-11 operating system is equipped with a playback device for the Sunstrand cartridge. A program on the PDP-11 accesses the data on the cartridge, via direct memory access techniques, and writes the Sunstrand data to an unlabeled, 9-track, 1600 BPI magnetic tape. The data acquired from the Sunstrand cartridge include Sunstrand status information. Appendix C, Section II describes the format of the tape generated by the PDP-11.

Program REFORMAT, on the VAX, restructures the PDP-11 data tape into blocks that are easily transported to other computer systems. REFORMAT also verifies the correct placement of all sync marks in each major frame (see Appendix C, Section I) and eliminates incomplete or erroneous major frames when producing the output file. Appendix C, Section III details the revised Sunstrand data file blocking structure.

From this revised database format, the visual radiometer database is generated, according to the description furnished by Mazzella and Larson[,] using program VRDATA, as shown in Figure 4. Programs REFORMAT and VRDATA have been fully tested using sample data.

Section III - Data Merging

The final section of the VIPER Data Processing System is the merging of the aforementioned databases into a unified database with which analysis of zodiacal data will be possible.

The first step in generating the unified database is to determine the center of the FOV and orientation of the VIPER cameras and the FOV for the visual radiometer. Program SFMATCHING matches the VCR data that was transported to the VAX in the file STARS.CLS as described at the end of Section II, Part A with the coalesced stars and neighbors. The coalesced stars and neighbors are the ISAM databases from Section I that are referred to as the coalesced and proximal catalogues.

The processing, which is detailed in Appendix D, begins with the initialization of default data values and the acquisition of user processing options. The options specify the tolerances to use on determining matches, threshold values, and the maximum scan rate of the VIPER GAS Can due to the shuttle motion. A frame of VCR data is acquired from the STARS.CLS file. The stars/clusters are sorted by increasing magnitude. The brightest star is chosen to be the reference star and the relative distances are calculated for all other sources in the frame. Data from the NASA CAS tape, which consist of times and associated pointing information, are acquired

 Mazzella, A. and Larson, K., (30 July 1990), **Visual Photometric Experiment Data Analysis**, Scientific Report No. 4, **GL-TR-90-0222, page 9 and Appendix D, ADA231915**

to provide an initial estimate of the camera pointing. The tape is read until consecutive records are found with times that bracket the VCR frame's time. Linear interpolation of pointing information is performed using the data in the bracketing times' records unless an exact match is found.

A keyed access is performed on the coalesced catalogue to acquire a star with a magnitude within the specified range of the reference star. The keyed access provides a possible match source (PM — see Section I) for which neighbors are acquired from the proximal catalogue. If the neighbors of the reference star have magnitudes and relative distances that match the PMS's neighbors, within the tolerances provided by the user's processing options, the matching for this VCR frame is complete. Camera pointing information and orientation can be determined using the algorithm described in Mazzella and Larson'. If the PMS and its neighbors do not match, a new PMS is acquired from the coalesced catalogue and neighbors are acquired from the proximal catalogue. The new PMS is tested against the reference star field. If no matches can be found for the video frame's reference star, that reference is eliminated from consideration and the next brightest source in the frame is used as the reference star. Reference elimination continues until a termination condition of either a match is found, two failures to find a match, or no more references exist in the video frame.

When either of the termination conditions occur, the results of this frame's processing are reported. If a match is found, the video frame's time and the camera pointing information, described in Appendix E, Section I, are written out to the POINTING.DAT output data file. If no match is found, the video frame's time, the brightest star's magnitude, and the NASA CAS tape's estimate of the camera pointing are reported.

Video information will not be available for the later times in the VIPER data collection interval. When such is the case, suitable dummy values will be used as fillers on the "no match" output records. Once the processing results for a video frame are reported, subsequent frames are acquired and similarly processed until no more video frames exist in the STARS.CLS input file.

The last step in the VIPER Data Processing System involves the merging of the pointing data in the MATCHING.DAT file with the visual radiometer database described in Section II, Part B. The user provides processing options that allow for time sub-ranges and tolerances to be selected in addition to providing visual radiometer pointing directions in shuttle body coordinates. The option of limiting processing to either pointing file data (from

---

' Mazzella, A. and Larson K., (30 July 1990), Visual Photometric Experiment Data Analysis, Scientific Report No. 4, GL-TR-90-0222, Appendices A and B, ADA231915

MATCHING.DAT), or NASA CAS data, is provided for evaluating the usefulness of these data sets based on the reliability and availability of the two pointing sources.

Program SPATIAL begins by acquiring the first VR data sample that falls within the time range set by the user processing options. If pointing data exist and star field referencing is a selected processing option, the appropriate data are acquired from MATCHING.DAT and tested for validity. Valid pointing data are consecutive data records that bracket, in time, the VR data sample's time such that the pointing data times are within the time tolerance of the VR data sample's time. Time tolerance is a user processing option. Linear interpolation is performed to determine celestial coordinates for the VR data sample. The bracketing pointing data's camera identifications are used to determine the value of a pointing source flag included in the output described in Appendix E, Section II.

If NASA CAS data exist for this time range and CAS referencing is a selected processing option, appropriate data are acquired from the NASA CAS database and tested for the same validity constraints as the pointing data. If the data are valid, VR pointing coordinates are calculated from the CAS data and the output flag identifying the sources used in the pointing determination is set accordingly. VR pointing coordinates determined from the star field take precedence over values found via CAS database values. The offset between CAS and VR coordinates is expected to be a constant (but initially unknown) transformation. The determination of this transformation will be based on a comparison of the CAS and video camera attitude information for the initial period of the flight operations, when both data sources are available.

If no VR pointing coordinate values can be found, fill-in values are placed in the output fields. This event can happen when bracketing time values either do not exist, fail the validation criterion, or are disabled via processing options. Before the next VR data sample is processed, results of the current sample are stored in the output buffer, to be written out as described in Appendix E, Section II. A detailed processing description for program SPATIAL is provided in Appendix F.

Figure 5 provides a flowchart description of the data merging process of the VIPER Data Processing System.

Two items of significance are pending before complete validation of the software can be achieved. First, VR and video camera calibration have not been performed. These items were pending successful completion of test data generation. Second, a program to create a CAS database of shuttle ephemeris values must be created.
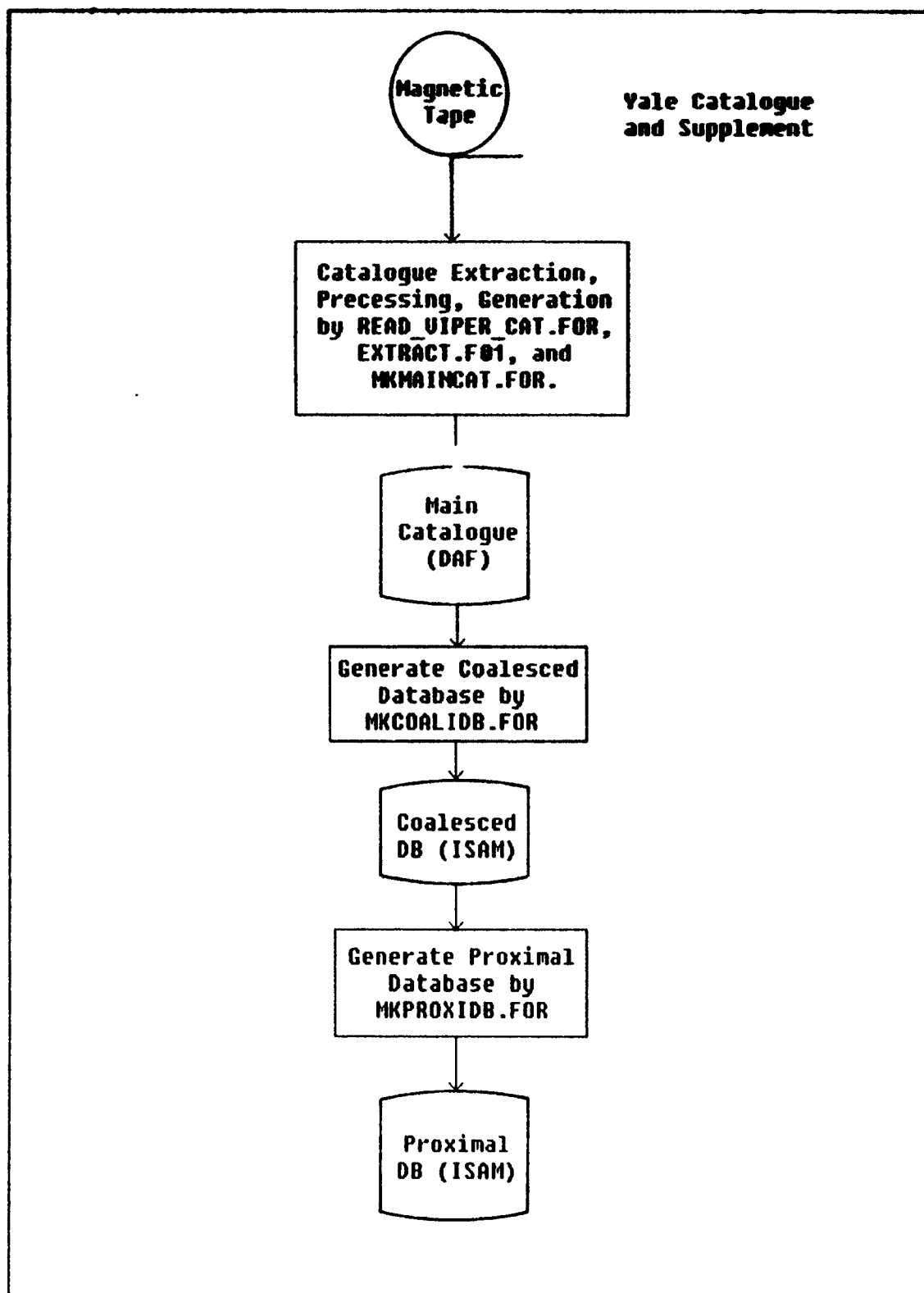
Figure 1 - Flowchart of Bright Star Catalogue Development

Figure 2 Sample Video Data Display with VR Field-Of-View
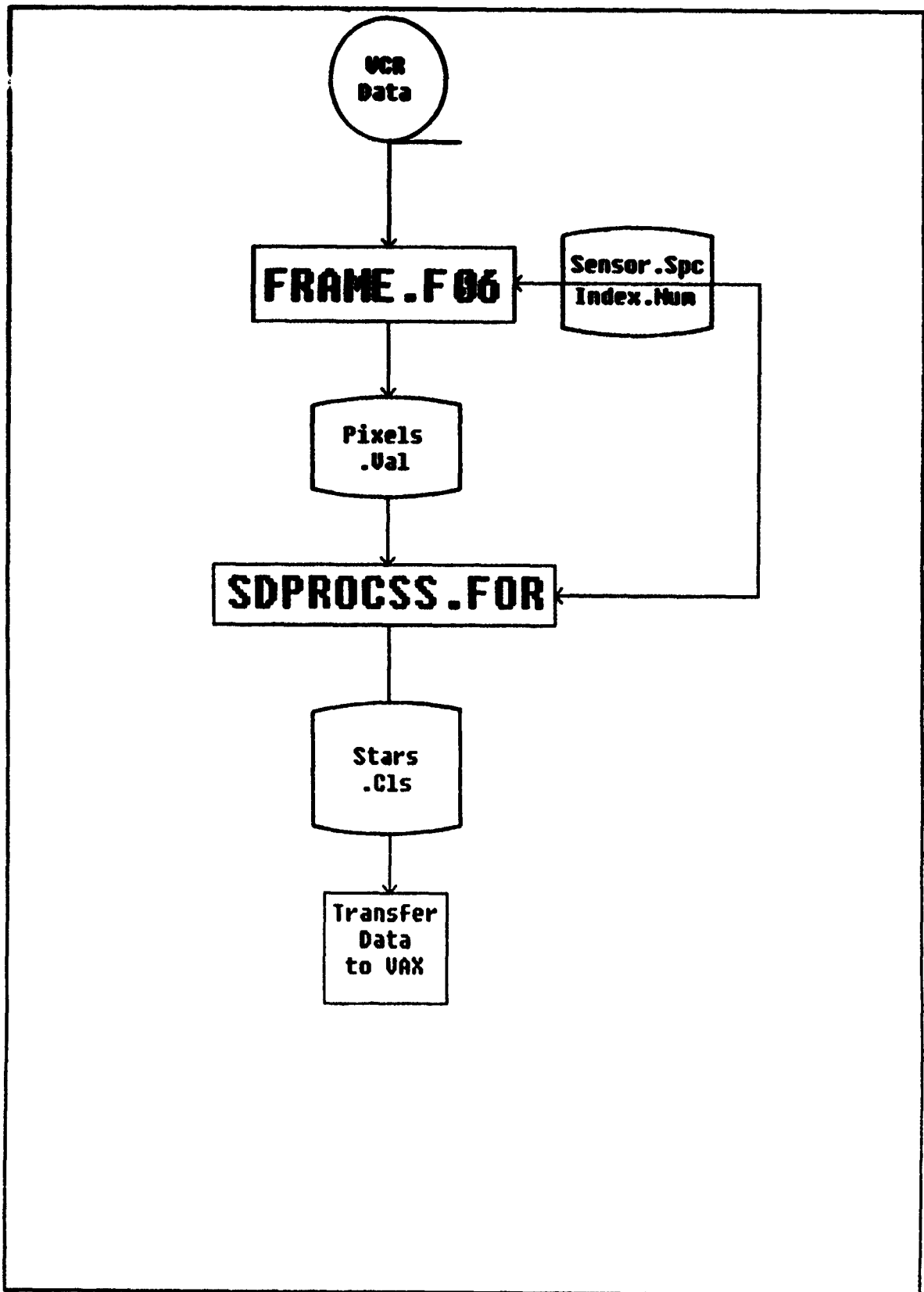
11

Figure 3 - Flowchart of Data Processing Steps on Zenith Z248
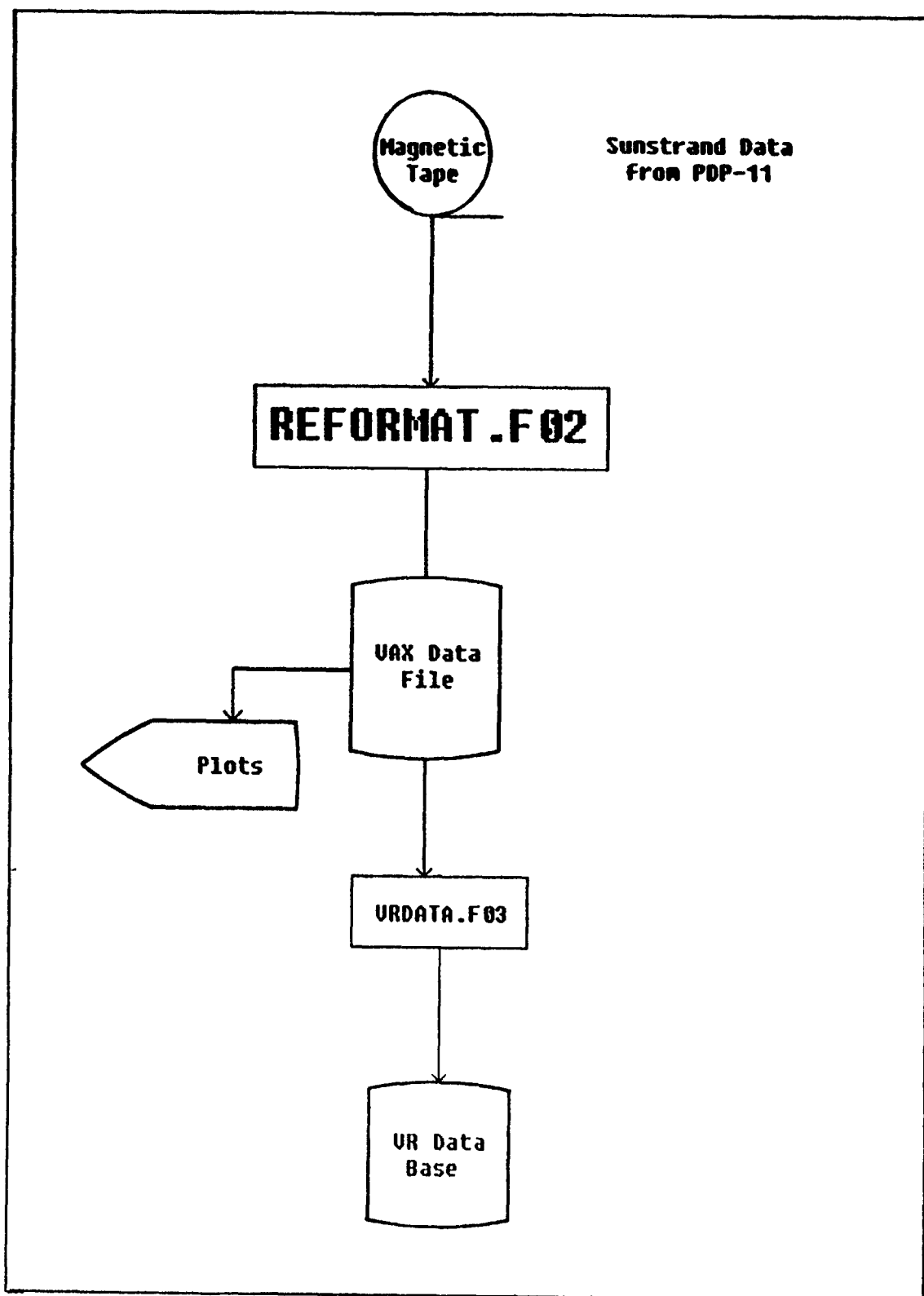
12

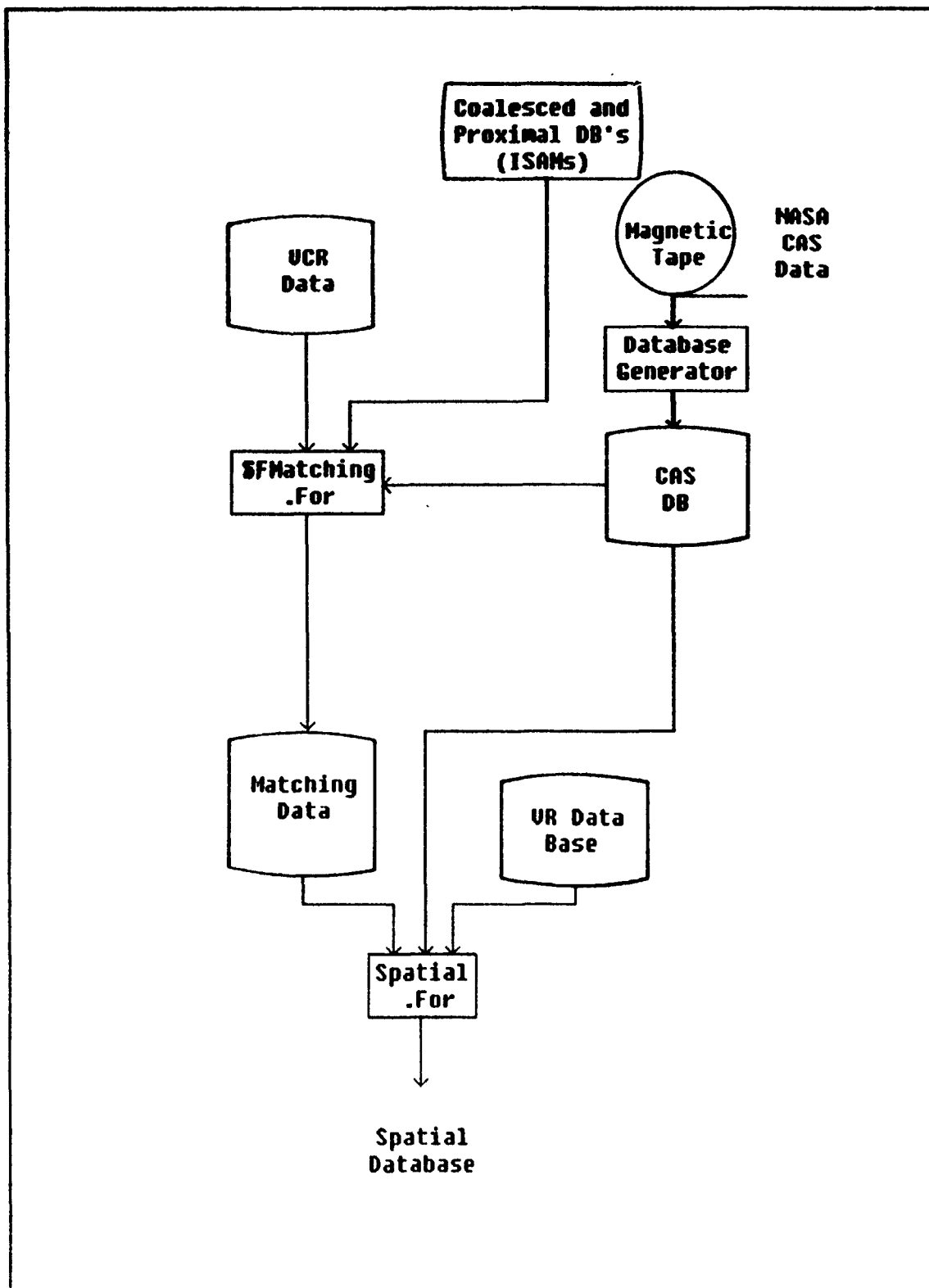**Figure 4** - Flowchart of Visual Radiometer Database Generation

Figure 5 - Flowchart of Data Merging to Form VIPER Spatial Database

14

Glossary of Acronyms

BCD – Binary Coded Decimal. A means of coding a decimal value using four-bit bytes.

BPI – Bits Per Inch. A measurement of the density at which data are written on a tape.

CNN – Current Neighbor's Number. The number refers to the ISAM catalogue ordinal number associated with a star. This catalogue number references the current neighbor's information in the Bright Star Catalogue.

DAF – Direct Access File. A file organizational scheme used in FORTRAN.

DEC – Digital Equipment Corporation. Manufacturer of the VAX computer system.

DOS – Disk Operating System. The most prevalent operating system on IBM personal computers.

FOV – Field-Of-View.

FORTRAN – FORmula TRANslator. A programming language commonly used for scientific applications.

GAS Can – Get-Away Special Canister used to carry payloads in the shuttle bay.

IBM – International Business Machines Corporation.

ISAM – Indexed Sequential Access Method. A file organizational scheme used in VAX/VMS FORTRAN.

PL/GP – Phillips Laboratory/Geophysics Directorate. The organizational name for the Air Force installation where this mission was initially conceived.

PMS – Possible Match Source.

RSN – Reference Star's Number. The number refers to the ISAM catalogue ordinal number associated with a star. This catalogue number references this video frame's reference star's information in the Bright Star Catalogue.

VCR – Video Cassette Recorder.

VHS – A format for storing analog video images on a VCR's cassette tapes.

VIPER – VIsual Photometric ExpeRiment. The acronym given to this mission.

VR – Visual Radiometer. An photosensitive detector in the VIPER package. The data gathered is in the visual region of the spectrum.

VRDATA – A program name as well as a format description of visual radiometer data. The program and format are detailed in Mazzella and Larson .

List of Programs

READ_VIPER_CAT.FOR

A VAX/VMS program to read the Yale Bright Star Catalogue and Supplement to generate a reformatted version containing only the data needed for the VIPER star discovery processing.

EXTRACT.F01

A VAX/VMS program to read the output from READ_VIPER_CAT.FOR and precess the stars according to their proper motion to their location on the date of data gathering. The stars are also tested against a magnitude threshold so that only the brightest sources are retained.

MKMAINCAT.FOR

A VAX/VMS program to read the output from EXTRACT.F01 and generate the DAF catalogue.

MKCOALIDB.FOR

A VAX/VMS program that reads the DAF catalogue from MKMAINCAT.FOR and generates the Coalesced Catalogue ISAM described in Appendix A, Section I. This catalogue is discussed in Section I of the report.

MKPROXIDB.FOR

A VAX/VMS program that generates the Proximal Catalogue ISAM described in Appendix A, Section II. This catalogue is discussed in Section I of the report.

FRAME.F06

A Zenith-PC program that acquires a single frame of video data from the VCR tape. The output from this program is the file PIXELS.VAL described in Appendix B, Section III.

SDPROCSS.FOR

A Zenith-PC program that reads the PIXELS.VAL file created by FRAME.F06, performs the star discovery processing, and writes the file STARS.CLS described in Appendix B, Section III.

REFORMAT.FOR

A VAX/VMS program that reads the VIPER Data Stream from magnetic tape and reformats the data for easier use on the VAX by subsequent programs. See Appendix C for data format specifics.

VRDATA.F03

A VAX/VMS program that reads the reformatted VIPER Data Stream and generates the VR database.

SFMATCHING.F01

A VAX/VMS program that matches the star fields recorded on the VCR tape and written (eventually) to file STARS.CLS to matching information in the Yale Bright Star Catalogues, i.e. coalesced and proximal catalogues.

SPATIAL.F01

A VAX/VMS program that merges the pointing information determined in the SFMATCHING.F01 program with the VR data in the VR database.

16

APPENDIX A

Appendix A

Section I
Coalesced Catalogue
File Structure

The Coalesced Catalogue is an ISAM data file created by the
VAX/Record Management System (RMS) utility. It consists of
fixed-length records of 12 bytes (3 longwords). Two fields on
each record are defined to be keys for Indexed (or keyed) access.
Both keys are two bytes of type INTEGER. The first key is the
reference star's number (RSN). The second key is the coded
magnitude with uncertainty key defined according to Equation 1 of
Section I of the main report. The remaining 8 bytes (2
longwords) are evenly divided between the celestial coordinate
values of right ascension and declination. The celestial
coordinates are of FORTRAN type REAL.

The following diagram illustrates the fields on each data record
in the Coalesced Catalogue.

16 bits	16 bits	32 bits	32 bits
Key #0 Catalogue Number (RSN) Integer	Key #1 Coded Magnitude Integer	Right Ascension (hours) Real	Declination (degrees) Real

Appendix A

Section II
Proximal Catalogue
File Structure

The Proximal Catalogue is an ISAM data file created by the VAX/RMS Utility with only one key for accessing the data. It has fixed-length records of 8 bytes (2 longwords). The key to each record is four bytes of type INTEGER defined according to Equation 2 of Section I of the main report. The remaining 4 bytes constitute another value of type INTEGER.

Records are grouped together to form a set of associated neighbors. For each entry in the Coalesced Catalogue, there is a set of associated neighbor records in the Proximal Catalogue, assuming that the same magnitude threshold conditions are used during both catalogues' generation. A set contains at least one record. The following diagram exemplifies how a set of neighbor records would be organized in the catalogue.

32 bits	32 bits	
Key #0 RSN*100,000	Number of Neighbors (n)	Header Record CNN = 0
Key #0 RSN*100,000 + CNN1	Radial Distance	Neighbor Record 1
Key #0 RSN*100,000 + CNN2	Radial Distance	Neighbor Record 2
. . .	. . .	
Key #0 RSN*100,000 + CNNn	Radial Distance	Neighbor Record "n"

The number of neighbors (n) may, in theory, equal 0 but in practice it does not because of the camera sensitivities and large fields-of-view and the cutoff magnitude for the catalogue.

Appendix A

Section III
CAS Data Items

The format for the CAS data generated by NASA is selectable by
the requestor in that a large set of data is available and the
requestor can chose any subset desired. A minimum set of
parameters has been defined for utilization with the VIPER
pointing determination programs. A minimum sampling rate of 0.5
Hz is required for those periods for which the visual radiometer
is recording data. (The standard CAS data is generated at 1.0
Hertz.)

The following is the format and description of the values needed
by the pointing programs.

Field 1:	Orbit Number
Fields 2-7:	Greenwich Mean Time (Orbiter): year, month, day, hour, minute, second
Fields 8-13:	Greenwich Mean Time (Mission Control): year, month, day, hour, minute, second
Fields 14, 15:	Position and velocity state vectors of orbiter: Aries true-of-date Cartesian coordinates (km; km/sec)
Fields 16-18:	Orbital elements, inclination (degrees) and right ascension of ascending node (degrees) of orbiter: Aries true-of-date coordinates
Field 19:	Sunrise (1)/sunset (0) flag
Field 20:	Attitude flag: 0 = telemetry; 1 = interpolation; 2 = attitude time line (inertial hold); 3 = attitude time line (solar inertial); 4 = attitude time line (local vertical/local horizontal); 5 = attitude time line (rotor); 6 = default attitude (Aries 1950 axes)
Fields 21-23:	Euler angles for aligning orbiter axes to UVW axes: yaw, pitch, roll (degrees)
Field 24:	Total angular attitude rate: Aries mean of 1950 coordinates (deg/sec)
Fields 25-27:	Earth position unit vector: shuttle body axis coordinates
Fields 28-30:	Position state vector of moon: Aries true-of-date coordinates (km)
Fields 31-33:	Position state vector of sun: Aries true-of-date coordinates (km)

APPENDIX B

21

Appendix B

Section I
VIPER Camera Specification File

Programs FRAME and SDPROCSS require calibration information for the VIPER cameras. This information is provided to the programs via the SENSOR.SPC file. Twelve values are given on a single record. The first two values are integer values for the intensity threshold. The intensity threshold is used to determine whether a pixel is part of a star. The remaining values are the critical distance, which defines two pixels as belonging to the same source, row and column scan rate components, a maximum intensity value, the uncertainty in the camera intensity, maximum time code intensity used to identify which camera is active, the identification number associated with the maximum time code intensity, minimum average pixel value for valid time codes, and time code thresholds that determine whether a bit is "on" for each camera.

The following is an example of the contents of a SENSOR.SPC file. The first line is the actual values acquired by the program. The second line is not acquired by the program. It contains the names of the variables in the FRAME and SDPROCSS programs into which the values are placed. Variables with more than one value are arrays containing values for the Xybion and Pulnix respectively.

80 80	7.	1.	1.	255.	170.	153	1	2000	120 120
In10Thrsh	DCrit	A	B	Lo	Del_L	MaxTC	MaxId	MinAve	IBThresh

Pulnix timecode is assumed brighter in this sample file. This configuration was needed for 15-Nov-1990 tests.

Appendix B

Section II
VIPER VCR Time Code
Bit Position Indexing File

Program FRAME acquires information from the file INDEX.NUM to properly decode the time value stored in row 15 of the VCR video frame. The information provided in this file directs FRAME to the pixels that constitute the binary-coded-decimal representation of the time. The pixel locations that are defined are compared to a threshold value. Values above the threshold are the binary digit 1. Values below the threshold are the binary digit 0.

The file contains the following information.

Variable Name	Description
BitWdth	Bit spacing measured as pixels per bit
Epsilon	Offset value used to fine-tune the time acquisition routine.
SyncSkp	Number of bits comprising the sync mark that need to be skipped. (SyncSkp*BitWdth)+1 is the first pixel in the time code.
NIndxNum	Number of binary digits in the time code. NIndxNum must always be 48 unless modifications are made to the source code in program FRAME.
IndxNum	Position of each of the NIndxNum bits. These are pixel offset values from the end of the sync mark. Eight of these index values are on each of the six lines in the INDEX.NUM file.

A total of ten records are in a standard INDEX.NUM file. The values that were found to produce the best results on the various laboratory test VCR tapes are shown in the following lines. Note the structure in comparison to the above descriptions.

```
  3.287
   .0000000
  11.000
 48
    3.287    6.574    9.861   13.148   16.435   19.722   23.009   26.296
   29.583   32.870   36.157   39.444   42.731   46.018   49.305   52.592
   55.879   59.166   62.453   65.740   69.027   72.314   75.601   78.888
   82.175   85.462   88.749   92.036   95.323   98.610  101.897  105.184
  108.471  111.758  115.045  118.332  121.619  124.906  128.193  131.480
  134.767  138.054  141.341  144.628  147.915  151.202  154.489  157.776
```

23

Appendix B

Section III
VIPER Pixel Value and Star Cluster File Formats

The VIPER PC programs FRAME and SDPROCSS perform data acquisition from the VIPER video tape and determine the weighted center of star clusters on the video image. The intermediate file, PIXELS.VAL, and final data file, STARS.CLS, have similar formats (though the meaning of the data is different). The file will contain "n+2" records per frame. For the PIXELS.VAL file, the meaning of "n" is the number of pixels found to have values above the threshold. For the STARS.CLS file, "n" refers to the number of stars seen in the video frame.

A diagram of the file's contents appears below. Please note that the record number listed with the data below does not appear in the file. It is given here to make discussion of the data easier.

Record Number	Data Description	Example
1	Date, Time, Camera Id	10/21 20:30:26.09 1
2 Thru n	Column Number, Row Number, Frame Grabber Intensity (PIXELS.VAL)	210 176 204
	Or	
2 Thru n	Column Number, Row Number, Magnitude of each Star Cluster (for STARS.CLS)	210 176 4.38
n+1	End-Of-Frame Flag	-999 -999 -999

APPENDIX C

Section I
VIPER Format

Column widths: columns 1–30 are 8 bits each; columns 19–20 form a 12‑bit field.

MF	1	2–3	4	5–6	7	8–9	10	11	12	13–14	15	16	17	18	19–20	21	22	23	24	25–26	27	28	29	30
1	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	FA	F3	20	A48	A49	D2	D3	D4	A48	A49	A18	EB	90
2	A48	A49	A21	A9	A48	A49	A14	A16	A17	A48	A49	A0	M2	SP8	A48	A49	D5	D6	D7	A48	A49	A19	EB	90
3	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A1	A31	SP8	A48	A49	D8	D9	D10	A48	A49	A18	EB	90
4	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A2	M2	SP8	A48	A49	D11	D12	D13	A48	A49	A19	EB	90
5	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A3	A32	SP8	A48	A49	D14	D15	D16	A48	A49	A18	EB	90
6	A48	A49	A21	01	A48	A49	A14	A16	A17	A48	A49	A4	M2	SP8	A48	A49	D17	D18	D19	A48	A49	A19	EB	90
7	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A5	A33	SP8	A48	A49	D20	D21	D22	A48	A49	A18	EB	90
8	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A6	A34	SP8	A48	A49	D23	D24	D25	A48	A49	A19	EB	90
9	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A7	M2	SP8	A48	A49	D26	SP8	SP8	A48	A49	A18	EB	90
10	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A8	A35	SP8	A48	A49	SP8	SP8	SP8	A48	A49	A19	EB	90
11	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A10	A36	SP8	A48	A49	D2	D3	D4	A48	A49	A18	EB	90
12	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A22	M2	SP8	A48	A49	D5	D6	D7	A48	A49	A19	EB	90
13	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A23	A37	SP8	A48	A49	D8	D9	D10	A48	A49	A18	EB	90
14	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A24	M2	SP8	A48	A49	D11	D12	D13	A48	A49	A19	EB	90
15	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A25	M1	SP8	A48	A49	D14	D15	D16	A48	A49	A18	EB	90
16	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A26	M2	SP8	A48	A49	D17	D18	D19	A48	A49	A19	EB	90
17	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A27	M3	SP8	A48	A49	D20	D21	D22	A48	A49	A18	EB	90
18	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A28	SP8	SP8	A48	A49	D23	D24	D25	A48	A49	A19	EB	90
19	A48	A49	A15	A20	A48	A49	A11	A12	A13	A48	A49	A29	SP8	SP8	A48	A49	D26	SP8	SP8	A48	A49	A18	EB	90
20	A48	A49	A21	SP12	A48	A49	A14	A16	A17	A48	A49	A30	SP8	SP8	A48	A49	SP8	SP8	SP8	A48	A49	A19	EB	90

One major frame is acquired for each second of data. A major frame is subdivided into 20 minor frames. Each minor frame is terminated by the hex value EB90, called the minor sync mark, in the last 16 bits. The hex value FAF320 in the first minor frame is the major sync mark which indicates that a new major frame has begun.

Appendix C

VIPER Designations

Input Desig	VIPER INPUT FUNCTION	# per sec	Input Range	Bits/ Word	Word #	Frame #	Remarks
A0	Solar + Press Transducer power bus 1	1	0-5V	8	16	1	
A1	Vis. Rad. (VR) pwr bus 2	1	0-5V	8	17	1	
A2	Motor pwr bus 3	1	0-5V	8	18	1	
A3	Video pwr bus 4	1	0-5V	8	16	2	
A4	Shutter/Filt Whl CTR pwr bus 5	1	0-5V	8	17	2	
A5	Video Tape 12V bus 4a	1	0-5V	8	18	2	
A6	Camera 12V bus 4b	1	0-5V	8	16	3	
A7	EXP +15V bus 5a	1	0-5V	8	17	3	
A8	EXP -15V bus 5b	1	0-5V	8	18	3	
A9	EXP +5V bus 5c	1	0-10V	12	4	1	
A10	EXP -5V bus 5d	1	0-5V	8	16	4	
A11	Shutter Position Status	10	0-5V	8	10	Even	
A12	Shutter Drive Motor	10	0-5V	8	11	Even	
A13	Shutter Control	10	0-5V	8	12	Even	
A14	Shutter Command	10	0-5V	8	10	Odd	
A15	Shutter Position Pot	10	0-10V	12	3	Even	
A16	Filter Position Status	10	0-5V	8	11	Odd	
A17	Filter Drive Motor	10	0-5V	8	12	Odd	
A18	Filter Control	10	0-5V	8	28	Even	
A19	Filter Command	10	0-5V	8	28	Odd	
A20	Filter Position Pot Monitor	10	0-10V	12	4	Even	
A21	Filter Position Pot Control	10	0-10V	12	3	Odd	
A22	Sun Present	1	0-5V	8	17	4	
A23	Sun Sensor Analog A	1	0-10V	8	18	4	
A24	Sun Sensor Analog B	1	0-10V	8	16	5	
A25	Camera #1 Iris Position	1	0-5V	8	17	5	
A26	Camera #2 Iris Position	1	0-5V	8	18	5	
A27	Camera Select	1	0-5V	8	16	6	
A28	Ambient Pressure	1	0-5V	8	17	6	
A29	Temp Cal Volt	1	0-5V	8	18	6	
A30	Temp Sensor T-1	1	0-5V	8	16	7	
A31	Temp Sensor T-2	1	0-5V	8	17	7	
A32	Temp Sensor T-3	1	0-5V	8	18	7	
A33	Temp Sensor T-4	1	0-5V	8	16	8	
A34	Temp Sensor T-5	1	0-5V	8	17	8	
A35	Temp Sensor T-6	1	0-5V	8	18	8	
A36	Analog Current x10 (CTR)	1	0-5V	8	16	9	
A37	Analog Current x100 (CTR)	1	0-5V	8	17	9	
A48	VR Hi Gain	100	0-10V	12	1,5,9, 13,17	All	SUPERCOM
A49	VR Lo Gain	100	0-10V	12	2,6,10 14,18	All	SUPERCOM

Appendix C

VIPER Designations

Input Desig	VIPER INPUT FUNCTION	# per sec	Input Range	Bits/ Word	Word #	Frame #	Remarks
D1	Shaft Encoder (S.E.)	1	Dig	12	4	3	
D2	Temp 1 (CTR) Battery	2	Dig	8	16	0,10	
D3	Temp 2 (CTR) VR IPS	2	Dig	8	17	0,10	
D4	Temp 3 (CTR) Xybion	2	Dig	8	18	0,10	
D5	Temp 4 (CTR) Pulnix	2	Dig	8	16	1,11	
D6	Temp 5 (CTR) Top Plate	2	Dig	8	17	1,11	
D7	Temp 6 (CTR) VHS Recorder	2	Dig	8	18	1,11	
D8	Temp 7 (CTR) Controller	2	Dig	8	16	2,12	
D9	Temp 8 (CTR) VR Electronics	2	Dig	8	17	2,12	
D10	Temp 9 (CTR) Tape Recorder	2	Dig	8	18	2,12	
D11	Temp 10 (CTR) Encoder	2	Dig	8	16	3,13	
D12	Temp 11 (CTR) PowerCube Cnvtr	2	Dig	8	17	3,13	
D13	Current Mon (CTR)	2	Dig	8	18	3,13	
D14	+5V Mon (CTR)	2	Dig	8	16	4,14	
D15	+15V Mon (CTR)	2	Dig	8	17	4,14	
D16	+28V Mon (CTR)	2	Dig	8	18	4,14	
D17	-15V Mon (CTR)	2	Dig	8	16	5,15	
D18	Clock 1/100 + 1/10 Sec (CTR)	2	Dig	8	17	5,15	
D19	Clock Sec (CTR)	2	Dig	8	18	5,15	
D20	Clock Minutes (CTR)	2	Dig	8	16	6,16	
D21	Clock Hours (CTR)	2	Dig	8	17	6,16	
D22	Clock Day of Month (CTR)	2	Dig	8	18	6,16	
D23	Clock Month (CTR)	2	Dig	8	16	7,17	
D24	Tape Status	2	Dig	8	17	7,17	
D25	Cover Status	2	Dig	8	18	7,17	
D26	Mode Status	2	Dig	8	16	8,18	
D29	PCM Status Word	2	Dig	8	17	18	
H1	Controller Handshake Monitor	1		8	17	16	
H2	Tape Recrdr Handshake Monitor	5		8	17	1,5,9, 13,17	
H3	PCM Status Monitor	1		8	17	18	

Section II

The data tape generated on the PDP-11 computer system contains alternating status and data blocks. A status block contains 3 words of information. For purposes of discussion, all computer "words" mentioned are assumed to be 16 bits in length. The first word contains the number of bytes in the status block. The next word provides the total length, in words, of the subsequent data block. The last word indicates the number of bytes of actual VIPER data in the subsequent data block. The difference between these last two values constitutes the extra status information added by the tape generating program. Before the termination of the tape generating program, a status block with zero values for the last two words is written to the tape. Because all data blocks are the same size, all status blocks, except the last one, have the following form.

Word 1	Word 2	Word 3
0006	4002	7FF0	(in hexadecimal)

The last status block appears as follows.

Word 1	Word 2	Word 3
0006	0000	0000	(in hexadecimal)

The data blocks are 16,387 (4002 in hexadecimal) words long. The first word indicates the number of bytes in this data block. This is followed by 32,752 (7FF0 in hexadecimal) bytes of actual VIPER data as recorded by the Sunstrand Data Logger. The data block ends with 20 bytes of status information from the Sunstrand Data Logger. The data blocks have the following form.

Word 1	Word 2	...	Word 7FF0	Word 7FF1	...	Word 8006	(in hexadecimal)
8006	dddd	...	dddd	ssss	...	80A0	(in hexadecimal)

"dddd" represents a hexadecimal VIPER data item and "ssss" represents a hexadecimal Sunstrand status item. All the data blocks end with the hexadecimal value of 80A0.

Investigation of VAX/FORTRAN Input/Output statements determined an allowable range of sizes for records from magnetic tapes as being

29

between 18 and 32767 bytes'. Both the status block, whose length
is 6 bytes, and the data block, whose length is 32774 bytes, are
outside of this range. Attempts to read these tape blocks using
ANSI FORTRAN techniques on the VAX computer system failed.

However, undocumented VAX VMS/FORTRAN extensions were discovered by
Real Time Engineering, Incorporated. These extensions were
employed in a reformatting program that writes the data to a file
in smaller block sizes that are easily transported among computer
systems. By mounting the tape as a "Foreign" tape without header-3
information and a block size of 65,000, a FORTRAN program can OPEN
the tape as an unformatted, variable-length-record file. By adding
a dummy 16-bit integer variable to the beginning of the I/O list on
each READ statement, the full amount of data in each block can be
acquired even though the FORTRAN Language manual states a limiting
range exists.

VAX FORTRAN, Volume 2, Language Reference Manual, June 1988,
digital equipment corporation, Maynard, Massachusetts; for VMS
Version 5.0 or higher and FORTRAN Version 5.0; page 9-9

Appendix C

Section III

The VAX data tape generated by program REFORMAT has block sizes
that enable easy data acquisition on virtually any computer system.
The tape is a 1600 BPI magnetic tape with fixed-length records of
binary data. Each record is 18000 bytes long and contains exactly
30 major frames of VIPER data. A single record constitutes a
single tape block. If program REFORMAT finds anything amiss in the
data (i.e. missing sync marks or incomplete major frames), 600
bytes of zeroes (one major frame) are written to the output stream
and the corrupted input data frame is skipped.

APPENDIX D

Appendix D - VIPER Star Field Matching Processing Description

The following is a detailed description of the processing steps
of the Star Field Matching program. The pseudo-code given here
is the basis of the SFMATCHING program. Most of the items in
parenthesis are either the actual variable names used in the
program, names of files that are accessed, or are FORTRAN
variable types associated with items. Items that are enclosed in
curly-brackets (i.e. ()) are module names. Module names that are
underlined denote the beginning of the pseudo-code associated
with the name. A module name imbedded in a pseudo-code statement
is treated as a FORTRAN call to that module.

Input:
1) Star field list from PC star discovery processing, with stars
   sorted by increasing magnitude: (Filename: STARS.CLS)
        Frame time, Camera ID, Number of stars (NumFldStar);
        Pixel row (IYPixel), Pixel column (JXPixel), and Magnitude
            (FieldMag) for each star.

2) Star catalogue
   Binary search, by magnitude, is performed on Coalesced Star
   Catalogue (COAL.IDB); neighboring stars are acquired from
   Proximal Catalogue (PROX.IDB).
   Note: Coalesced Star Catalogue is assumed to be acquired
         completely into memory in magnitude-order.

3) CAS data estimate (see Appendix A Section II for details).

Process:

{ACQUIRE} Acquire comparative pointing positions:
    a) CAS tape result, for corresponding time;
    Is there a CAS tape result for this frame time (or an adjacent
       bracketing pair for interpolation)?
       Yes: Set condition to check index star versus CAS;
       No: Clear condition to check index star versus CAS;
    b) Previous star field result, if time difference is within
       limits (DeltaTFrame < DeltaTime...);

{SETFRAME} Set up star field frame:
    Reference "plate scale" conversions, using coordinates
       referenced to center of field, based on camera identifier:
       For i = 1 to NumFldStar
           DistX(i) = FactX * (XMidPixel - JXPixel(i))
           DistY(i) = FactY * (YMidPixel - IYPixel(i))
       Next i
    Define brightest star magnitude, FieldMag0:
    Initialize error flag for catalogue match failures for index
       star:

34

        IndexError = 0

Initialization test:
    Is time difference between current frame and previous frame
        (DeltaTFrame) less than (frame length)/(estimated scan
        rate)?
        ["Estimated scan rate" is nominally 1 degree per second,
        and will be defined as a program variable.]
        Yes: Is current FieldMag0 approximately equal to previous
            FieldMag0?
            Yes: Use proximal catalogue field for previous frame as
                initial reference for current frame; proceed to
                calculation of relative field star distances
                (RELDIST);
            No: Prepare to search index for new match to FieldMag0
                (FIRSTSTAR);
        No: Prepare to search index for new match to FieldMag0
            (FIRSTSTAR);

(FIRSTSTAR) Find index star with magnitude (CatMag) within
    tolerances of FieldMag0;
        Binary search for star in index catalogue:
        IndexMin = 0
        IndexMax = Number of index catalogue entries
        Set initial distance comparison counter limit:
            IniDist = 2
        (SEARCH)
            Index = (IndexMin + IndexMax)/2
            If CatMag(Index) = FieldMag0 Then
                Proceed to find first star in index catalogue with
                    this magnitude (FINDFIRST);
            Else If CatMag(Index) > FieldMag0 Then
                IndexMax = Index
            Else
                IndexMin = Index
            Endif
            If IndexMax > IndexMin Then
                Continue search process (SEARCH)
            Else
                [The field star magnitude occurs in a gap in the
                    catalogue magnitudes];
                Check the catalogue entry found against the field
                    star, with error tolerances:
                If $|CatMag(IndexMin) - FieldMag0| < TolMag$ Then
                    Set Index = IndexMin
                    Proceed to search for first index catalogue entry
                        with magnitude equal to CatMag(Index)
                        (FINDFIRST);
                Else If $|CatMag(IndexMin+1) - FieldMag0| < TolMag$
                Then

35

```
                    Set Index = IndexMin + 1
                    Proceed to search for first index catalogue entry
                        with magnitude equal to CatMag(Index)
                        (FINDFIRST); (it should be this entry, the
                        first catalogue magnitude above the gap);
                Else
                    Warning: Error finding index star!
                    Set FieldMag0 = FieldMag(2) (skip to next
                        brightest star in field);
                    Make the original index star "disappear":
                        FieldMag(1) = FieldMag(2)
                        DistX(1) = DistX(2)
                        DistY(1) = DistY(2)
                    Set initial distance comparison counter limit:
                        IniDist = 3
                    Increment error counter:
                        IndexError = IndexError + 1
                    If IndexError < 2 Then
                        Proceed to search for new reference star in
                            index catalogue (FIRSTSTAR);
                    Else
                        Report failure of frame match (FAILURE);
                        Proceed to next star field (ACQUIRE);
                    Endif
                Endif
            Endif
        Endif
```

(FINDFIRST) Find first star in catalogue with this magnitude:
```
    If Index > 1 Then
        If CatMag(Index-1) = CatMag(Index) Then
            Index = Index - 1
            Loop to check previous entry in catalogue (FINDFIRST)
        Endif
    Endif
    Set:
        IndexLow = Index
        IndexHigh = Index
        Advance = TRUE
        RefMag = CatMag(Index)
        CheckLimit = 0
        (for use in finding 'next' index candidate, if necessary);
```

(RELDIST) Define distances from brightest star:
```
    X0 = DistX(1)
    Y0 = DistY(1)
    For i = IniDist to NumFldStar
        FieldDist(i) = sqrt ((DistX(i) - X0)² + (DistY(i) - Y0)²)
    Next i
```

36

Appendix D - VIPER Star Field Matching Processing Description

{PATHCHECK} Preliainary check of brightest field star against CAS position;
    If CAS data is available Then
        Compare star position to CAS data;
        If discrepancy exceeds tc'erances Then
            Search for new index ..car {NEXTINDEX}
        Endif
    Endif

{CHECKFRAME} Compare proximal catalogue entries to field stars:
    Select subset from proximal catalogue for match to field stars:
        Ciear ProxUsed array (TRUE/FALSE flags, indicating that a proximal catalogue star has already been matched to a field star): ProxUsed(j) = FALSE, for all j;
        BadMatch = 0 (counter for number of unmatched field stars)
        For i = IniDist to NumFldStar
            For j = 1 to NumProx
                If ProxUsed(j) = TRUE Then Next j
                If |FieldDist(i)-ProxDist(j)| < TolDist Then
                    If |FieldMag(i)-ProxMag(j)| < TolMag
                        IndexProx(i) = j
                        ProxUsed(j) = TRUE
                        Next i
                  Endif
                Endif
            Next j
            Warning: Matching star not found! [Have completed 'j' loop with no match.]
            If i < i_{rw} Then
                [Star field shows early discrepancy from proximal catalogue];
                Search for new index star {NEXTINDEX}
            Else
                BadMatch = BadMatch + 1
                IndexProx(i) = 0 (flag field star as having no catalogue match);
                If BadMatch > $BadMatch_{rw}$ Then search for new index star {NEXTINDEX}
            Endif
         Next i

{POINTING} Determine field center, orientation, and visual radiometer pointing:
    Determine coordinate transformation from camera to celestial coordinates using least-square minimization;
    Can determine orientation about line-of-sight from one of the Euler angles for the transformation;
    Use transformation matrix to compute:

      Celestial coordinates for center of camera field;
      Celestial coordinates for center of visual radiometer,
         based on its camera coordinate offset;
    (See Mazzella and Larson for separate description for
    transformation determination.)
    If DeltaTFrame < DeltaTime$_{cn}$ Then
      Compute angular distance from previous position (DeltaAng);
      If DeltaAng/DeltaTFrame > Critical scan rate Then
        Warning: Excessive scan rate!
        Report results (REPORT);
      Endif
    Endif

(REPORT) Report results:
    Report (to pointing file):
      Frame time;
      Camera pointing (center) in celestial coordinates;
      Camera orientation, as azimuthal degrees from North about
        pointing direction;
      Root mean squared distance error for star field versus
        catalogue;
      Root mean squared intensity error for star field versus
        catalogue;
      Visual radiometer pointing in celestial coordinates;
      Number of stars in visual radiometer field;
      Estimated total intensity of stars in visual radiometer
        field;

(SAVEFRAME) Save parameters of current frame for reference by
    subsequent frame (only for successful field matches);
    Frame time;
    FieldMag0;
    Camera pointing position in celestial coordinates;
Proceed to next star field (ACQUIRE);

****
(NEXTINDEX) Procedure to select the next candidate from the index
    catalogue for the field reference star;
    If IndexHigh < number of stars in index catalogue Then
      If CatMag(Index+1) = RefMag Then
        IndexHigh = IndexHigh + 1
        Index = Index + 1
        Proceed to preliminary check for proximal catalogue
          match (PATHCHECK);
      Endif
    Else If Advance = TRUE Then
      Advance = FALSE
      If IndexHigh = Number of index catalogue entries Then
        CheckLimit = CheckLimit + 1

Appendix D - VIPER Star Field Matching Processing Description

```
            Loop on search for next index star (NEXTINDEX);
        Endif
        If |CatMag(IndexHigh+1) - FieldMag0| < TolMag Then
            IndexHigh = IndexHigh + 1
            Index = IndexHigh
            CheckLimit = 0
            Proceed to preliminary check for proximal catalogue
                match (PATHCHECK);
        Else
            CheckLimit = CheckLimit + 1
        Endif
    Else
        Advance = TRUE
        If IndexLow = 1 Then
            CheckLimit = CheckLimit + 1
            Loop on search for next index star (NEXTINDEX);
        Endif
        If |CatMag(IndexLow-1) - FieldMag0| < TolMag Then
            IndexLow = IndexLow - 1
            Index = IndexLow
            CheckLimit = 0
            Proceed to preliminary check for proximal catalogue
                match (PATHCHECK);
        Else
            CheckLimit = CheckLimit + 1
        Endif
    Endif
    If CheckLimit < 2 Then
        (at least one end of search is still within magnitude
            tolerances)
        Loop on search for next index star (NEXTINDEX);
    Else
        Warning: No further index star candidates!
        Set FieldMag0 = FieldMag(2) (skip to next brightest star in
            field);
        Make the original index star "disappear":
            FieldMag(1) = FieldMag(2)
            DistX(1) = DistX(2)
            DistY(1) = DistY(2)
        Set initial distance comparison counter limit:
            IniDist = 3
        Increment error counter:
            IndexError = IndexError + 1
        If IndexError < 2 Then
            Proceed to search for new reference star in index
                catalogue (FIRSTSTAR);
        Else
            Report failure of frame match (FAILURE);
            Proceed to next star field (ACQUIRE);
        Endif
```

Endif

****
(FAILURE) Report failure of star field match;
   Frame time;
   CAS tape estimate for pointing, if available;
   Magnitude of brightest field star;

APPENDIX E

41

Appendix E - File Format for Data Merging Programs

Section I - Output from the SFMATCHING program

The star field matching program, SFMATCHING, determines the attitude of the VIPER video cameras, and thereby the visual radiometer, by comparing a VCR video frame with known sources from the Yale Bright Star Catalogue. When a successful match occurs between the video frame and the catalogue, the following information is written to the ASCII file POINTING.DAT.

A. Time stamp of frame - Julian date, hours, minutes, and seconds
B. Camera Identification number
C. Estimated total intensity of stars in VR field
D. Camera pointing - Celestial coordinates of the center of the FOV of the video camera.
E. VR pointing - Celestial coordinates of the center of the FOV of the visual radiometer.
F. Camera Orientation - Euler rotation angles from an initially aligned camera/celestial coordinate system.

Appendix E - File Format for Data Merging Programs

Section II - Output from the SPATIAL program

The spatial data base generating program, SPATIAL, produces a data file that merges shuttle ephemeris data with the VR data values. The data are written out in variable length blocks with a maximum size of 3242 bytes. The first word (i.e. 2 bytes) contains the number of records in the block. Each record is 54 bytes in length and a maximum of 60 records is allowed for each block.

A.	VIPER day count since 1 January 1985.	4 bytes of INTEGER
B.	Time of day in seconds.	4 bytes of REAL
C.	Shutter position.	2 bytes of INTEGER
D.	Filter position.	2 bytes of INTEGER
E.	VR temperature, in counts.	2 bytes of INTEGER
F.	VR radiance, in counts.	2 bytes of INTEGER
G.	VR right ascension, in degrees.	4 bytes of REAL
H.	VR declination, in degrees.	4 bytes of REAL
I.	Shuttle right ascension, in degrees.	4 bytes of REAL
J.	Shuttle declination, in degrees.	4 bytes of REAL
K.	Shuttle altitude, in km.	4 bytes of REAL
L.	Solar right ascension, in degrees.	4 bytes of REAL
M.	Solar declination, in degrees.	4 bytes of REAL
N.	Lunar right ascension, in degrees.	4 bytes of REAL
O.	Lunar declination, in degrees.	4 bytes of REAL
P.	Source flag.	2 bytes of INTEGER

    0 - no position
    1 - Xybion only
    2 - Pulnix only
    3 - Xybion and Pulnix without CAS
    4 - CAS only
    5 - Xybion and CAS
    6 - Pulnix and CAS
    7 - Xybion, Pulnix, and CAS
Note: Absence of CAS data implies missing coordinates of sun and moon, and missing orbital parameters.

APPENDIX F

Appendix F - SPATIAL Program's Processing Description

The following is a detailed description of the processing steps
of the Spatial Data Base Generating program. The pseudo-code
given here is the basis of the SPATIAL program. Most of the
items in parenthesis are either the actual variable names used in
the program, names of files that are accessed, or are FORTRAN
variable types associated with items. Items that are enclosed in
curly-brackets (i.e. {}) are module names. Module names that are
underlined denote the beginning of the pseudo-code associated
with the name. A module name imbedded in a pseudo-code statement
is treated as a FORTRAN call to that module.

Inputs:
  1) VRDATA = Visual Radiometer data base, in VRDATA format;
  2) MATCHING.DAT = pointing data from star field matching;
  3) CAS data base = pointing transformation angles, derived from
     CAS data, with additional celestial parameters.

User Processing Options:
  1) Starting month/day/year/time for processing;
  2) Ending month/day/year/time for processing;
  3) Pointing directions (zenith: VRZEN, azimuth: VRAZ angles)
     for Visual Radiometer in shuttle body coordinates; [VRZEN =
     200 if no estimate for usage with CAS is available; default
     is VRZEN = 180, VRAZ = 0] (For shuttle body, X is forward, Y
     is over right wing, and Z is toward bottom.)
  4) Time tolerance (TIMEINT), in seconds, for bracketing
     pointing information relative to visual radiometer data time
     [default is ±7 seconds];
  5) POINTFLG = initial status for star field matching file
     (FALSE = no star field matching file);
  6) PATHFLG = initial status for CAS file (FALSE = no CAS file).

Processing:

{INIT} (Initialization):
Set PTSPEC(1) = PTSPEC(2) = 0 [Time specifications for star field
  pointing data.]
Set ATSPEC(1) = ATSPEC(2) = 0 [Time specifications for CAS data.]
Read input specifications [See "User Processing Options" above.]
and convert start and end specifications to VIPER Julian day and
fraction.

{VRINP}:
Read VRDATA items: VDATE (date: YYMMDD), VTIME (time: SSSSS.FFF),
  SHPOS (shutter position), FILPOS (filter position), VTEMP
  (temperature), NBUF (buffer size), VRRAD(N), N = 1, NBUF
  (radiance).
If end-of-file then Go to LASTBUF.
Merge VDATE, VTIME into day and fraction form: VTSPEC =

46

NDAYS((VDATE)) + VTIME(in seconds)/86400 (double precision).
If VTSPEC < start time then Go to VRINP. [could lose remainder
  of up to 1 second of data in block after start time.]
If VTSPEC > end time then Go to LASTBUF.
Begin loop VRSAMPL, for NSAMPL = 1 to NBUF:
Initialize visual radiometer pointing position, assuming no
  pointing information is available (can revise this pointing
  later):
  SPVRRA = 0;
  SPVRDEC = 100.
  SRCFLG = 0 (pointing source flag initialization).

[STARINP]:
  If POINTFLG = FALSE then Go to PATHINP.
  If PTSPEC(1) <= VTSPEC < PTSPEC(2) then Go to POINTING.
  (For case of VTSPEC outside PTSPEC bounds:)
  Read MATCHING.DAT items, searching for matching or bracketing
    time for VTSPEC:
    PDATE(I) (date: MMDD), PTIME(I) (time: HHMMSS.FF), VRRA(I)
      (VR right ascension, degrees), VRDEC(I) (VR declination,
      degrees), TVFLG(I) (camera), I = 1, 2.
  If end-of-file for I = 1, then
    Set POINTFLG = FALSE;
    SET PTSPEC(1) = PTSPEC(2) = 0;
    Go to PATHINP;
  Endif
  If end-of-file for I = 2, then
    Set PDATE(2) = PDATE(1), PTIME(2) = PTIME(1), VRRA(2) =
      VRRA(1), VRDEC(2) = VRDEC(1);
    Set POINTFLG = FALSE;
  Endif
  Merge PDATE, PTIME into day and fraction form, using year
    supplied in input specifications [This may be performed while
    searching for matching/bracketing values.]: PTSPEC(I) =
    NDAYS((PDATE(I))) + PTIME(I) (in seconds)/86400 (double
    precision); should have PTSPEC(1) <= VTSPEC < PTSPEC(2) [if
    VTSPEC > PTSPEC(2) due to EOF, will ignore values later].

[POINTING]:
  [Check that successive star field references are not so far
    apart as to be invalid references.]
  If PTSPEC(1) < (VTSPEC - TIMEINT/86400) or PTSPEC(2) > (VTSPEC
    + TIMEINT/86400) then Go to PATHINP;
  If PTSPEC(2) < VTSPEC then Go to PATHINP;
  If VTSPEC = PTSPEC(1) Then
    SPVRRA = VRRA(1); (Visual Radiometer right ascension)
    SPVRDEC = VRDEC(1); (Visual Radiometer declination)
    If TVFLG(1) = 0 [Pulnix] Then
      Set SRCFLG = 2;
    Else If TVFLG(1) = 1 [Xybion] Then

47

```
        Set SRCFLG = 1;
      Endif
   Else
      Interpolate between (VRRA(1),VRDEC(1)) and (VRRA(2),VRDEC(2))
        according to time intervals defined by PTSPEC, VTSPEC to
        obtain SPVRRA and SPVRDEC;
      If TVFLG(1) = TVFLG(2) = 0 [Pulnix] Then
        Set SRCFLG = 2;
      Else If TVFLG(1) = TVFLG(2) = 1 [Xybion] Then
        Set SRCFLG = 1;
      Else
        Set SRCFLG = 3; (both cameras used)
      Endif
   Endif
```

[PATHINP]:
```
   If PATHFLG = FALSE then Go to NOPATH.
   If ATSPEC(1) <= VTSPEC < ATSPEC(2) then Go to PATH.
   [For case of VTSPEC outside ATSPEC bounds.]
   Read CAS items, searching for matching or bracketing time for
     VTSPEC:
     ADATE(J,I) J = 1, 3 (date: year, month, day), ATIME(J,I) J =
       1, 3 (time: hour, minute, second), AEUL(J,I) J = 1, 3
       (Euler angles for shuttle orientation), APOS(J,I) J = 1, 2
       (right ascension/declination angles for shuttle position),
       ARAD(I) (radial position for shuttle), ORBINC(I)
       (inclination of shuttle orbit), ORBNODE(I) (right ascension
       of shuttle orbit ascending node), ASUN(J,I) J = 1, 2 (right
       ascension/declination angles for sun), AMOON(J,I) J = 1, 2
       (right ascension/declination angles for moon), I = 1, 2.
   If end-of-file for I = 1, then
     Set PATHFLG = FALSE;
     SET ATSPEC(1) = ATSPEC(2) = 0;
     Go to NOPATH;
   Endif
   If end-of-file for I = 2, then
     Set ADATE(J,2) = ADATE(J,1), ATIME(J,2) = ATIME(J,1),
       AEUL(J,2) = AEUL(J,1), APOS(J,2) = APOS(J,1), ARAD(2) =
       ARAD(1), ASUN(J,2) = ASUN(J,1), AMOON(J,2) = AMOON(J,1);
     Set PATHFLG = FALSE;
   Endif
   Merge ADATE, ATIME into day and fraction form (this may be
     performed while searching for matching/bracketing values):
     ATSPEC(I) = NDAYS((ADATE(J,I))) + ATIME(1,I)/24 +
     ATIME(2,I)/1440 + ATIME(3,I)/86400 (double precision); should
     have ATSPEC(1) <= VTSPEC < ATSPEC(2) [if VTSPEC > ATSPEC(2)
     due to EOF, will ignore values later].
```

[PATH]:
```
   [Check that successive CAS references are not so far apart as
```

to be invalid references.]

If ATSPEC(1) < (VTSPEC - TIMEINT/86400) or ATSPEC(2) > (VTSPEC + TIMEINT/86400) then Go to NOPATH;

If ATSPEC(2) < VTSPEC then Go to NOPATH (beyond end of CAS data);

If SRCFLG .ne. 0 then Go to CELEST2 (can skip pointing determination from CAS if available from star fields).

If VRZEN = 200 then Go to CELEST2 (can skip pointing determination from CAS if no body coordinate pointing reference is available).

If VTSPEC = ATSPEC(1) Then

> Set SINGLE = TRUE (to indicate that only first transformation is needed to specify visual radiometer pointing).
>
> Use ORBINC(1), ORBNODE(1), APOS(J,1), J = 1, 2, ARAD(1) to compute transformation matrix TUVW1 for transformation from shuttle UVW coordinates to celestial coordinates (Procedure UVWCEL).
>
> Use Euler angles AEUL(J,1), J = 1, 3 to compute transformation matrix TXYZ1 for transformation from shuttle body coordinates to UVW coordinates (Procedure XYZUVW).
>
> Compute pointing direction for visual radiometer in celestial coordinates, for time ATSPEC(1) (Procedure PATHVR).

[CELEST1]:

Else

> Set SINGLE = FALSE [Indicating that both transformations are needed to specify visual radiometer pointing.]
>
> Use ORBINC(1), ORBNODE(1), APOS(J,1), J = 1, 2, to compute transformation matrix TUVW1 for transformation from shuttle UVW coordinates to celestial coordinates (Procedure UVWCEL).
>
> [Set a flag to determine whether recalculation of the transformation matrix TXYZ1 is required, based on new Euler angle inputs. Also, test the time to determine if possibility exists of using an earlier TXYZ2 for this transformation.]
>
> Use Euler angles AEUL(J,1), J = 1, 3 to compute transformation matrix TXYZ1 for transformation from shuttle body coordinates to UVW coordinates (Procedure XYZUVW).
>
> Compute pointing direction for visual radiometer in celestial coordinates, for time ATSPEC(1) (Procedure PATHVR).
>
> Use ORBINC(2), ORBNODE(2), APOS(J,2), J = 1, 2, to compute transformation matrix TUVW2 for transformation from shuttle UVW coordinates to celestial coordinates (Procedure UVWCEL).
>
> [Set a flag to determine whether recalculation of the transformation matrix TXYZ2 is required, based on new Euler angle inputs.]
>
> Use Euler angles AEUL(J,2), J = 1, 3 to compute transformation matrix TXYZ2 for transformation from shuttle body coordinates to UVW coordinates (Procedure XYZUVW).

Appendix F - SPATIAL Program's Processing Description

    Compute pointing direction for visual radiometer in celestial
      coordinates, for time ATSPEC(2) (Procedure PATHVR).
    Interpolate pointing directions for visual radiometer at
      times ATSPEC(1) and ATSPEC(2) to obtain visual radiometer
      pointing at time VTSPEC.
  Endif

(CELEST2):
[Store celestial object and shuttle reference positions, after
  interpolating between reference times ATSPEC(1) and ATSPEC(2),
  if necessary.]
  (SUNRA, SUNDEC) = Interp(ASUN, ATSPEC, VTSPEC, SINGLE)
  (LUNRA, LUNDEC) = Interp(AMOON, ATSPEC, VTSPEC, SINGLE)
  SRCFLG = SRCFLG + 4 (retain camera attribute; indicate CAS data
    is present).
  Go to STOREREC.

(NOPATH): (set fill-in values)
  ORBALT = -1;
  ORBRA = SUNRA = LUNRA = 0;
  ORBDEC = SUNDEC = LUNDEC = 100.

(STOREREC):
  Store record in output buffer.
  Increment NRECS.
  If NRECS = LIMBUF then
    Write output data block.
    Set NRECS = 0
  Endif
  Increment time for visual radiometer sample: VTSPEC = VTSPEC +
    0.01/86400.
  Next NSAMPL.
End VRSAMPL loop.
Stop.

(LASTBUF):
If NRECS .ne. 0 then Write output data block (NRECS <= LIMBUF).
Stop.

(UVWCEL):
A = APOS(1,I) (shuttle position right ascension);
D = APOS(2,I) (shuttle position declination);
K = ORBINC(I) (shuttle orbit inclination) (REAL);
N = ORBNODE(I) (shuttle orbit ascending node) (REAL);

U = (U1, U2, U3) = (cos A cos D, sin A cos D, sin D);
W = (W1, W2, W3) = (sin K sin N, -sin K cos N, cos K);
V = (V1, V2, V3) = cross-prod(W, U) (vector product);

$$TUVWI = \begin{bmatrix} U1 & V1 & W1 \\ U2 & V2 & W2 \\ U3 & V3 & W3 \end{bmatrix}$$

Return TUVWI.

(XYZUVW):
A = AEUL(1,I) (shuttle yaw Euler angle);
B = AEUL(2,I) (shuttle pitch Euler angle);
P = AEUL(3,I) (shuttle roll Euler angle);

$$TX = \begin{bmatrix} 1 & 0 & 0 \\ 0 & COS\ P & SIN\ P \\ 0 & -SIN\ P & COS\ P \end{bmatrix}$$

$$TY = \begin{bmatrix} COS\ B & 0 & -SIN\ B \\ 0 & 1 & 0 \\ SIN\ B & 0 & COS\ B \end{bmatrix}$$

$$TZ = \begin{bmatrix} COS\ A & SIN\ A & 0 \\ -SIN\ A & COS\ A & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

TXYZI = TX * TY * TZ
Return TXYZI.

(PATHVR):
A = VRAZ (VR azimuth, in shuttle coordinates);
Z = VRZEN (VR zenith, in shuttle coordinates); [nominally 180 degrees]

L = (cos A cos Z, sin A cos Z, sin Z);
[The above three steps are only required for the initial call for a given data processing run, because VRZEN and VRAZ remain fixed.]
SPVRI = TUVWI * TXYZI * L;
[Decompose SPVRI to right ascension and declination as follows:
  SPVRRAI = atan2(SPVRI2, SPVRI1) (degrees);
  SPVRDECI = acos (SPVRI3) (degrees).]
Return.

51